


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#)^{New!} [more »](#)

object oriented compiler optimization

Search

[Advanced Search](#)
[Preferences](#)
WebResults 1 - 10 of about 60,500 for object oriented compiler optimization. (0.37 seconds)**Comp.compilers: New TR: "Whole-Program Optimization of Object ...**

From comp.compilers newsgroup: New TR: Whole-Program **Optimization of Object-Oriented** Languages. New TR: "Whole-Program **Optimization of compilers.iecc.com/comparch/article/96-06-052** - 5k - [Cached](#) - [Similar pages](#)

Comp.compilers: Re: Object-oriented compiler construction: ideas?

... ie, CSE-as-an-object isn't very **object oriented**). ... I'll use virtual function calls on the **object-classes** instead ... for **optimization** and wants the **compiler** to bind ... [compilers.iecc.com/comparch/article/96-06-137](#) - 5k - [Cached](#) - [Similar pages](#)
[\[More results from compilers.iecc.com \]](#)

Advanced Constructs and Compiler Optimizations for a Parallel ...

"Advanced Constructs and **Compiler Optimizations** for a Parallel, **Object Oriented**, Shared Memory Language running on a distributed System", (zipped postscript ... [www.icsi.berkeley.edu/~sather/ Publications/fleiner.html](#) - 5k - [Cached](#) - [Similar pages](#)

Whole-Program Optimization of Object-Oriented Languages - Dean ...

... Languages - Dean, Chambers, Grove (1995) (Correct) Active bibliography (related documents): More All 1.7: Vortex: An **Optimizing Compiler for Object-Oriented**.. ... [citeseer.ist.psu.edu/136007.html](#) - 36k - [Cached](#) - [Similar pages](#)

Citations: Wholeprogram optimization of object-oriented languages ...

... 2. 4 Related Work This thesis is largely based on combining the work that has been done on **compiler optimizations** for **object oriented** languages [1, 28, 27, 28 ... [citeseer.ist.psu.edu/context/39171/166361](#) - 13k - [Cached](#) - [Similar pages](#)
[\[More results from citeseer.ist.psu.edu \]](#)

Vortex compiler

Vortex is a language-independent **optimizing compiler** infrastructure for **object-oriented** and other high-level languages, written entirely in Cecil. ...

[www.cs.washington.edu/research/ projects/cecil/www/vortex.html](#) - 9k - Apr 2, 2004 - [Cached](#) - [Similar pages](#)

Cecil/Vortex Project Paper: "Vortex: An Optimizing Compiler for ...

Vortex: An **Optimizing Compiler** for **Object-Oriented** Languages. Jeffrey Dean, Greg DeFouw, David Grove, Vass Litvinov, and Craig Chambers ...

[www.cs.washington.edu/research/projects/ cecil/www/Papers/hybrid.html](#) - 3k - Apr 2, 2004 - [Cached](#) - [Similar pages](#)
[\[More results from www.cs.washington.edu \]](#)

Object-oriented systems at UCSB

Our research group investigates **compiler optimizations** for **object-oriented** languages, related implementation aspects (such as dispatch mechanisms or garbage ...

[www.cs.ucsb.edu/labs/oocsb/](#) - 4k - [Cached](#) - [Similar pages](#)

Analysis and Compilation of Object-Oriented Languages

... Jeffrey Dean, Greg DeFouw, David Grove, Vassily Litvinov, and Craig Chambers.

Vortex: An **Optimizing Compiler** for **Object-Oriented** Languages. In OOPSLA'96. ...

[user.it.uu.se/~svenolof/wpo/oo-compilation-papers.html](#) - 9k - [Cached](#) - [Similar pages](#)

The Object-Oriented Numerics Page

... The Portland Group C++ **compiler**; The MPC++, a ... C++) and fe.lib (an **object-oriented** finite element ... Includes interval arithmetic, global **optimization**, and solving ...



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide

+"dynamic binding" +(optimize optimization optimizing)

SEARCH

THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **dynamic binding optimize optimization optimizing**Found **285** of **798,953**

Sort results by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)

Display results

expanded form

[Search Tips](#)[Try this search in The Digital Library](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐1 [An enabling optimization for C++ virtual functions](#)

Bradley M. Kuhn, David W. Binkley

February 1996 **Proceedings of the 1996 ACM symposium on Applied Computing**Full text available: [pdf\(655.81 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)**Keywords:** dynamic finding, optimization, virtual functions2 [Automatic prototype generating via optimized object model](#)

Sheldon X. Liang, Lynn Zhang, Luqi

June 2003 **ACM SIGAda Ada Letters**, Volume XXIII Issue 2Full text available: [pdf\(809.96 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Computer-aided prototyping shows promise that one system under development frees designers from implementation details by executing specifications via reusable components. Ada is first choice for constructing such reusable object-oriented components because Ada95 is the only international standard programming language that supports object-oriented real-time distributed systems. But Ada has diversified object forms that are so intricate that people feel it difficult to find an equivalence of a cl ...

Keywords: Automatic Prototype Generating (APG), compositional pattern, encapsulation, inheritance, polymorphism3 [Interactive type analysis and extended message splitting; optimizing dynamically-typed object-oriented programs](#)

Craig Chambers, David Ungar

June 1990 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation**, Volume 25 Issue 6Full text available: [pdf\(1.58 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Object-oriented languages have suffered from poor performance caused by frequent and slow dynamically-bound procedure calls. The best way to speed up a procedure call is to compile it out, but dynamic binding of object-oriented procedure calls without static receiver type information precludes inlining. Iterative type analysis and extended message splitting are new compilation techniques that extract much of the necessary type information and make it possible ...



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide

+abstract:constant +abstract:method +abstract:table

SEARCH

THE GUIDE TO COMPUTING LITERATURE


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **constant method table**Found **26** of **798,953**Sort results
by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Display
results

expanded form

[Search Tips](#)Try this search in [The Digital Library](#)
☐ Open results in a new
window

Results 1 - 20 of 26

Result page: **1** [2](#) [next](#)Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Optimizing multi-method dispatch using compressed dispatch tables](#)

Eric Amiel, Olivier Gruber, Eric Simon

 October 1994 **ACM SIGPLAN Notices , Proceedings of the ninth annual conference on
Object-oriented programming systems, language, and applications**, Volume
29 Issue 10

Full text available: pdf(1.83 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Optimizing method dispatch is a central issue in object-oriented language implementation. The dispatch table scheme, used for example by C++, is the only implementation of method dispatch that offers constant time performance. This property is the main asset of dispatch tables and a major requirement for some languages. However, the major drawback of dispatch tables is the space they require. Reducing the size of dispatch tables has been studied in the case ...

2 [Hash table methods for case statements](#)

Jason Gait

 April 1982 **Proceedings of the 20th annual Southeast regional conference**

Full text available: pdf(315.27 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#)

The CASE statement evaluates an expression, selects an action according to the value of the expression and then executes the action. The most efficient runtime behavior is exhibited when the action can be selected via a jump table, which provides an entry for every possible value in the range of the expression but has execution time that is constant as the number of cases increases. If the range of the expression is too large then the jump table becomes impractical because of excessive space req ...

Keywords: Case Statements, Compiler Implementation, Computer Architecture, Hashing, Open Addressing

3 [Optimal weight assignment for signature generation](#)

Chun-Wu Roger Leng, Dik Lun Lee

 June 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 2

Full text available: pdf(1.52 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Previous work on superimposed coding has been characterized by two aspects. First, it is generally assumed that signatures are generated from logical text blocks of the same size; that is, each block contains the same number of unique terms after stopword and duplicate removal. We call this approach the fixed-size block (FSB) method, since each text block has the same size, as measured by the number of unique terms contained in it. Second, with